TABLE OF CONTENTS
SECTION 2.0

SECTION 2.0 GETTING ACQUAINTED WITH SIGLAB 1

GOALS: 1
INTRODUCTION 1
EXAMPLE 1: SINGLE-CHANNEL MEASUREMENTS 1
Acquiring Data 4
A Simple Exercise to Build On 13
POST PROCESSING 18
Brief Architecture Overview 18
POST-PROCESSING WITH MATLAB 19
A Brief SLm example 20
Post-Processing from a Setup File 23
Two-Channel Measurements 24
POST-PROCESSING OUTSIDE OF MATLAB 25
A BRIEF REVIEW OF THE VNA SETUP WINDOW 26
Channel Dialog 26
MC Setup for Multi-Channel Users 28
Frequency Dialog 29
Processing Dialog 30
Triggering Dialog 31

SECTION 2.0
GETTING ACQUAINTED WITH SIGLAB

Goals:

- Learn the basic controls of the dynamic signal analyzer and function generator.
- Make measurements showing SigLab's excellent dynamic range and spectral purity.
- Extract the "just" acquired data and post-process in MATLAB.

Introduction

Most experts would agree. The best way to learn is by example. The hardest part of learning any new product is mastering the user interface. This section will accelerate you up the learning curve and have you taking full advantage of the SigLab user interface. So just relax and follow the instructions. By the time you finish this section, you'll have a good feel for what SigLab can do for you and just as importantly, what it cannot.

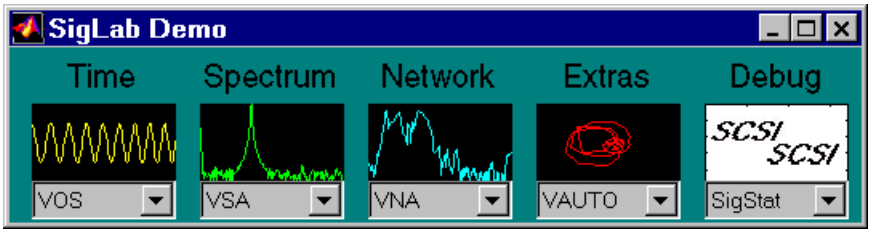
This section introduces two commonly used Vis (virtual instruments), the SigLab network analyzer and the function generator. There are many other VIs which you may find useful in your work and we encourage you to use these as well. In any case, the goals we outlined above are relevant regardless of the instrument. You must always set up your instrument, acquire data, and finally post-process the data to make inferences and generate reports. This section assumes that you have successfully completed the SigLab and SCSI installation.

Example 1: Single-Channel Measurements

1. **Turn on SigLab** and verify its connection to the PC. The front panel Status should briefly flash **OK**, and then change to **T5** (Terminated at SCSI ID 5) or similar. If you do not see this, make sure you have completed the SCSI Installation per the *Quick Start Guide.*
2. **Connect channel 1 output directly to channel 1 input** using the RCA cables and connectors provided with SigLab. Or you may use one of your own BNC cables.
3. **Start MATLAB**.
4. **Start SigDemo**. Invoke "SigDemo" from the MATLAB Command Window.

>> `sigdemo` <enter>

Note that MATLAB is case-sensitive in general but typing Sigdemo or even SIGDEMO will work under later versions of MATLAB 5.



5. **Start *vna***. From *SigLab Demo*, under Network, select ***vna*** (virtual network analyzer).

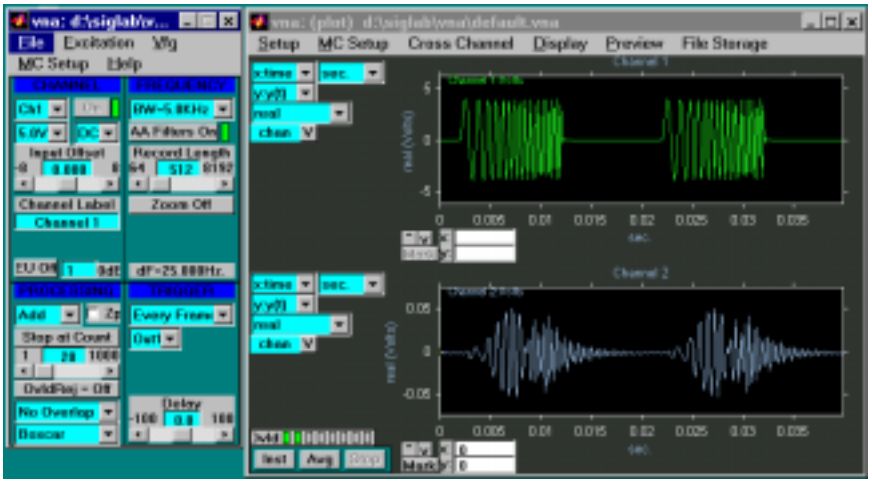


*Figure 2-1. The **vna** setup window (left) and the **vna** plot window (right) are both initialized here using the default.**vna** file.*

After some time, you should see the something very similar to Fig 2-1. appear.

8 Input Channels	Pentium 90 MHz	Pentium II 350 MHz
Pre-hardware Init	28 seconds	10 seconds
Post-hardware Init	11 seconds	2 seconds

Table 2-1. The time required to initialize SigLab instruments depends on whether the hardware has been pre-initialized.

On a 90 MHz Pentium computer, approximately 28 seconds is required to initialize ***vna*** for 8 channels. The first session requires more time since neither the software nor the hardware was initialized. Once you have initialized the hardware, subsequent invocations of ***vna*** will be substantially faster.

6. **Start vfg**. We will use the function generator as our source of excitation. Bring up the function generator by clicking on the **vfg** (virtual function generator) menu item in the ***vna*** setup window. You'll find **vfg** next to the File and Excitation menus. This menu item

behaves like a pushbutton in the sense that no submenu items appear but instead **vfg** is invoked directly.

7. **Importance of a Well-Organized Workspace**. Before we get into pushing buttons and acquiring data, the most important thing to do is establish an efficient working space. Otherwise, you'll spend more time mousing and keyboarding around and less time doing real work. Setting the workspace is a highly individual thing but there is one feature that most people can agree upon … **the more visibility, the better**. See Figure 2-2 for one particular way to organize your workspace. Resize the ***vna*** plot window and MATLAB window if necessary. We suggest you maintain some visibility with the MATLAB workspace, even if it's only a few lines for debugging purposes. Your screen resolution is the biggest factor limiting what organizational scheme you choose. The minimum recommended is 1024x768.

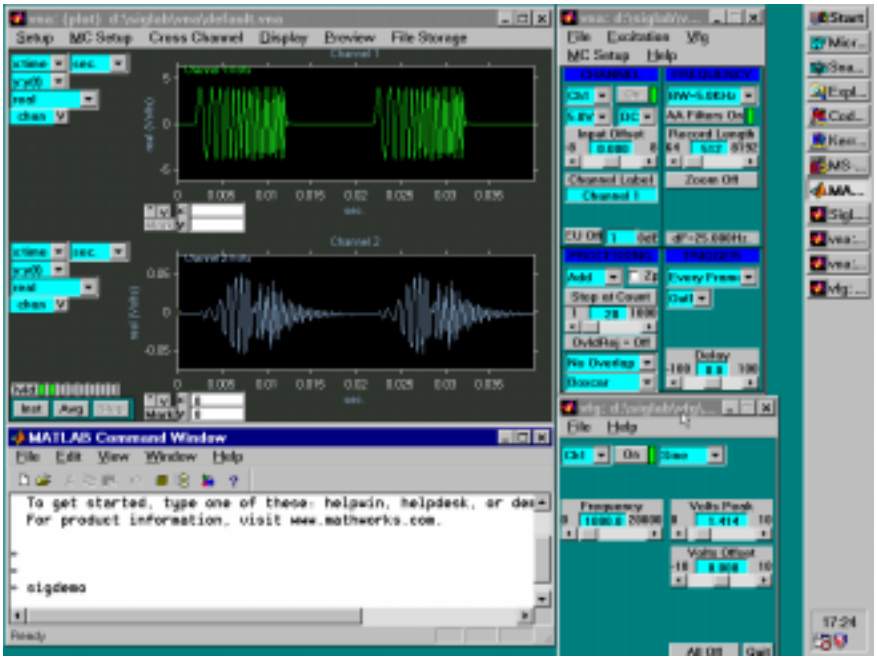
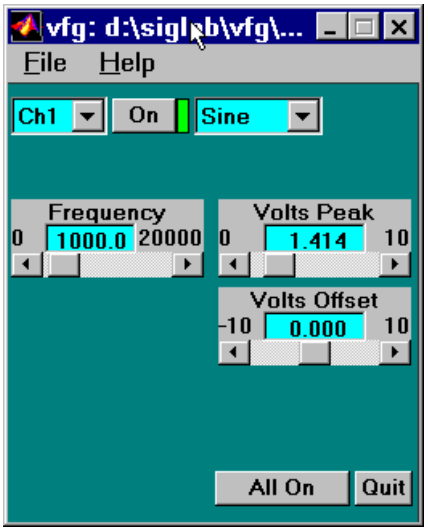


*Figure 2-2. One possible workspace organization for easy access to the **vna** plot window, **vna** setup window, **vfg** window, and the MATLAB Command Window.*

8. **Where did the data come from?** In the ***vna*** plot window, noticed that plots appear. You may wonder where that data came from since we haven't actually acquired any data yet. This plot is data from a file called *default.vna*, a name you'll see listed in the title bar. We call *default.vna* a setup file. Setup files contain data, the screen position and setup parameters as well like voltage ranges, sample rate, triggering, coupling, averaging mode, etc. All of this information is stored in one concise setup file for easy recall. Setup files make good memories far less important.

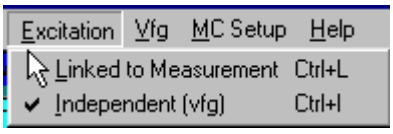
9. **vfg Window** The **vfg** is your knob to controlling SigLab's output channels. It lets you generate different waveforms for excitation of the device under test in real-time using ***vna***. Currently, you should have a connection from SigLab output channel 1 to Siglab input channel 1. Let's take a closer look at the **vfg** settings.



Currently, ***vfg*** is set to generate a 1.414 Volts Peak, 0 volts offset, 1000 Hz sine wave on channel 1. You should set the **All On/All Off** toggle button to read **All On** as in Figure 2-3. If your settings appear as in Figure 2-3, your SigLab is now outputting a 1000 Hz 1.414 volts peak sine wave.

10. **Excitation: Independent**. In the ***vna*** setup window, under the Excitation menu, ensures that the Independent selection is checked.

Figure 2-3. The default VFG window is set to a 1000 Hz sine wave on channel one.



Independent gives you full use of vfg.

Acquiring Data

This section lets you get hands-on experience with the instrument. You will be able to generate, acquire and display real data. A brief overview of every control will be covered in the following sections. For in-depth coverage of each control in ***vna*** and ***vfg***, refer to **Sections** 5.0 and 6.0 of this manual.

To get started, switch to the ***vna*** plot window and hit the **Inst** button (bottom left-hand corner). **Inst** stands for Instantaneous, which means no averages are to be taken.

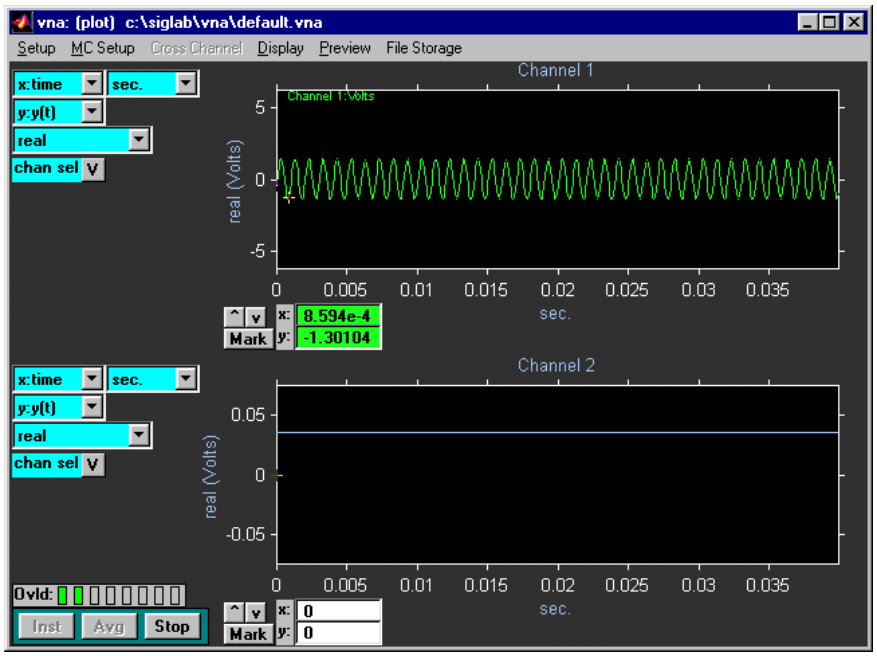
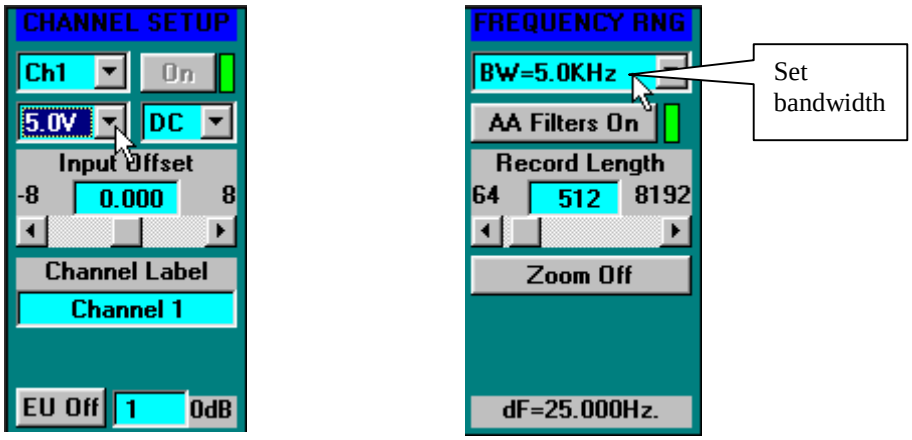


*Figure 2-4. The **vna** plot window appears as above after hitting the **Inst** button to acquire and display the sine wave generated by SigLab output channel 1. Notice that while you are acquiring, the Inst and Avg buttons are greyed out (disabled) while the adjacent Stop button is black (enabled).*

The plots should begin to update with newly acquired data. The top plot represents channel 1 time data, displayed in green. The bottom plot represents channel 2, displayed in time blue. You should see a flat line on channel 2 because we are not injecting a signal.

Making Changes On-the-Fly. One of SigLab's primary strengths is the ability to react to your changing demands on the fly. You can modify almost every control in the user interface while SigLab is acquiring data. Some people say that it feels very analog. To get a feel for this real-time interaction, do the following while it's acquiring data:



1. **Changing Bandwidth**. Change the bandwidth from 5 kHz to 10 kHz … to 20 kHz via the Frequency Rng dialog in the ***vna*** setup window. This popup has values from 20 kHz down to 2 Hz for a 20 kHz SigLab. Change it back to 5 kHz.

2. **Changing Full Scale Range**. Change the full scale voltage level on channel 1 from 5.0 volts to 10.0 volts … to .63 volts via the Channel Setup dialog in the ***vna*** setup window. This popup has values from 10.0 volts down to 20 mv. Notice that you saturated the analog front-end circuitry of SigLab when you changed to 0.63 volts. The waveform

saturated at 0.63 volts and the overload indicators turned red. See Figure 2-5. **Change** back to 5.0 volts.

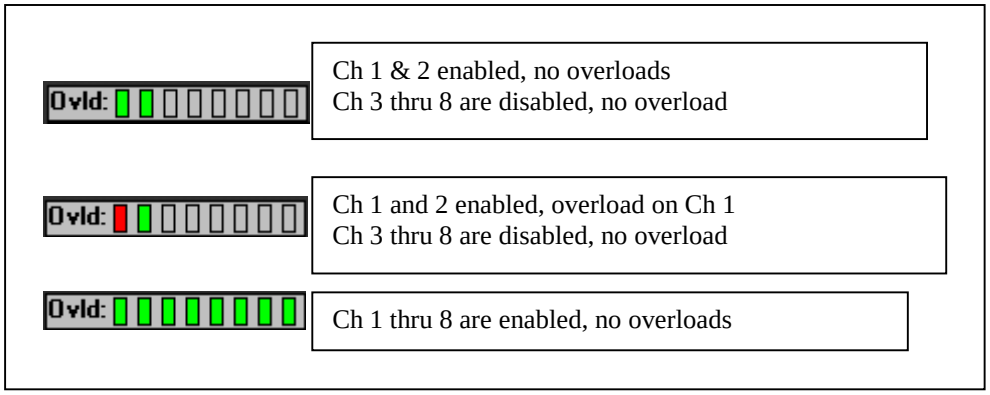
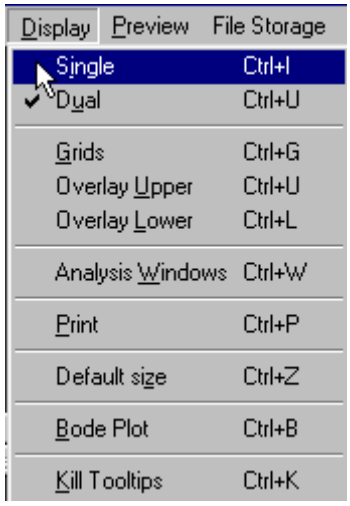

Ovld:
Ch 1 & 2 enabled, no overloads
Ch 3 thru 8 are disabled, no overload
Ovld:
Ch 1 and 2 enabled, overload on Ch 1
Ch 3 thru 8 are disabled, no overload
Ovld:
Ch 1 thru 8 are enabled, no overloads

*Figure 2-5. The SigLab overload (Ovld) indicators appear above the Inst, Avg, and Stop buttons in the **vna** plot window. Here are 3 possible states of the many you could see the overload indicators in for an 8 channel system.*

Some Facts about Overload Monitoring

- Green indicates channel is enabled with no overloads.
- Red indicates channel is enabled with an overload.
- Clear indicates channel is not enabled with no overloads.
- Yellow indicates that the channel is not enabled (data is not being acquired or processed on this channel) BUT, there is an overload condition. The overload condition should be addressed by either removing the input signal or increasing the full-scale range.
- There is one overload indicator per input channel. Channel 1's overload indicator is on the far left and they number sequentially from left to right. If you have a 16 channel system, you will see 16 overload indicators.
- Overload monitoring occurs before any filtering takes place. You may not see obvious saturated data in the plot window if the frequency components causing the overload are being heavily attenuated by the analog and/or digital alias filters. Use the overload indicators as your ultimate judge for overloaded conditions, not the plot.
- The front panel of the SigLab hardware also monitors and displays overloads. This LED turns red if an overload occurs on any input channel.



.

3. **Changing Display Mode**. Change to single-display in the ***vna*** plot window using the menu selection **Display/Single.** You should see one large plot of a sine wave.

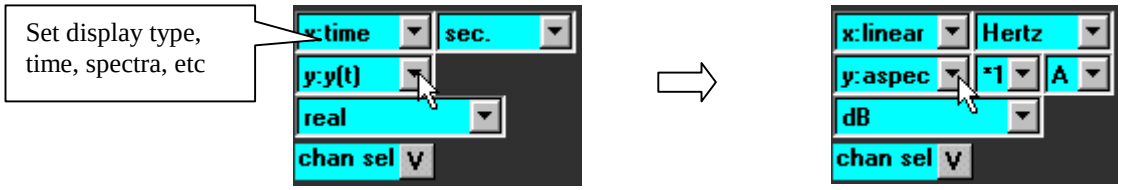


Figure 2-6. Changing the display from time to autospectrum

4. **Viewing Spectrum**. Change the data-type from **time** to **autospectrum** using the **y:** data type popup in the upper-left hand corner of the ***vna*** plot window. You should see the plot change from the time domain to the frequency domain.

   You should see a 0 dB spectral peak at 1 kHz. ***vna*** displays the autospectrum (aspec) of the acquired data. The autospectrum computation is implemented on SigLab's C31 DSP. Because we have a “pure” sine wave, we do not see significant harmonics or spurious components in the autospectrum.

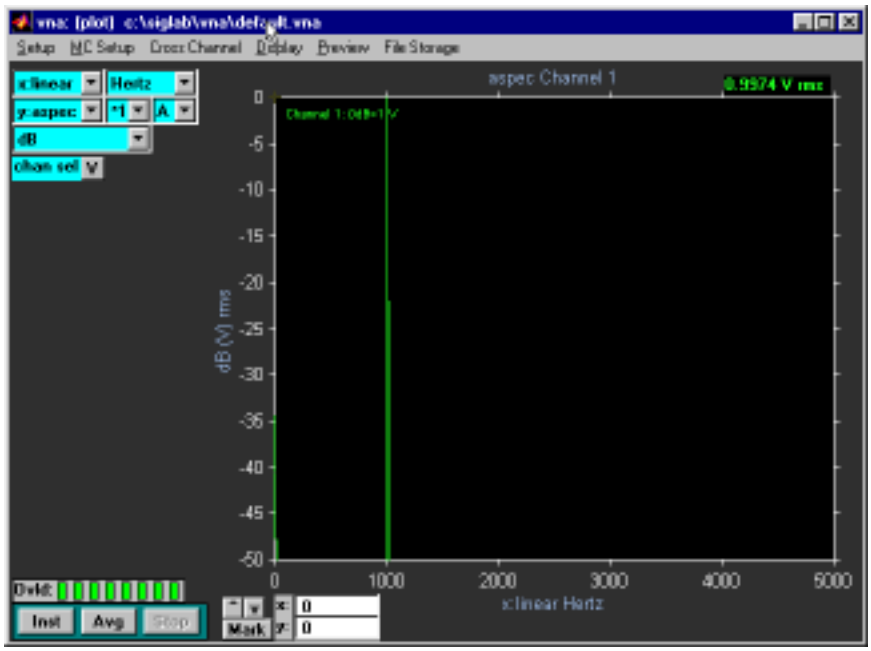


Figure 2-7. The Single Display mode after changing from time to autospectrum. Notice the plot is not autoscaled initially.

5. **Pre-Autoscaled**. You probably noticed that the noise floor of the sine wave is not initially visible. Let's fix that. Right single click anywhere in the plot window. Continue right clicking until you see the noise floor appear. Right single clicking is a way to **autoscale** the plot limits to the data.
6. **Post-Autoscaled**. After autoscaling you should see something like Figure 2-8.

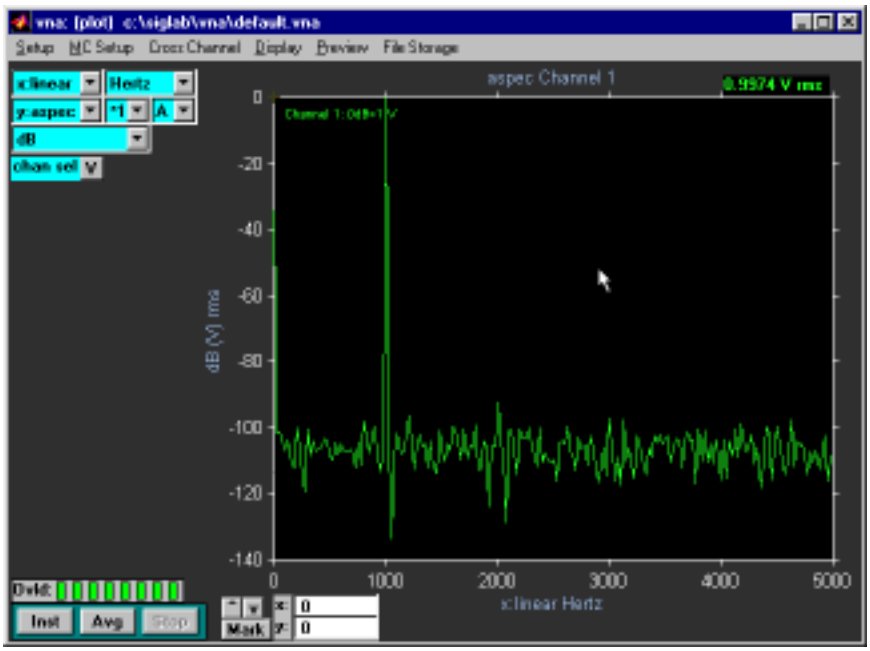


Figure 2-8. After autoscaling you can see the signal and the noise floor across the entire spectrum. Notice the excellent dynamic range attained even without the use of averaging.

7. **Left single click on the Stop button** in the ***vna*** plot window before proceeding with steps 8 through 14.

8. **Display Expansion**. At times it's necessary to zoom on parts of the spectrum for a closer look. Here's how to do this:

   a) **Position the mouse cursor** in the plot window to start the upper left portion of your zoom box.

   b) **Left double click and hold** after the second click.

   c) While holding down, **drag the mouse** down and to the right. A yellow box will appear as illustrated in Figure x.

   d) **Enclose the region of interest** in the yellow zoom box and then release the mouse.

   e) **Left single click** again in the plot window to accept the new plot limits.

   Repeat the above process until you have it mastered. Don't get frustrated if you don't get the knack right away. It's not easy at first to get the hang of display expansion. In one sentence it's: **left double-click, hold, drag, release, left single click to accept or right click to reject.**

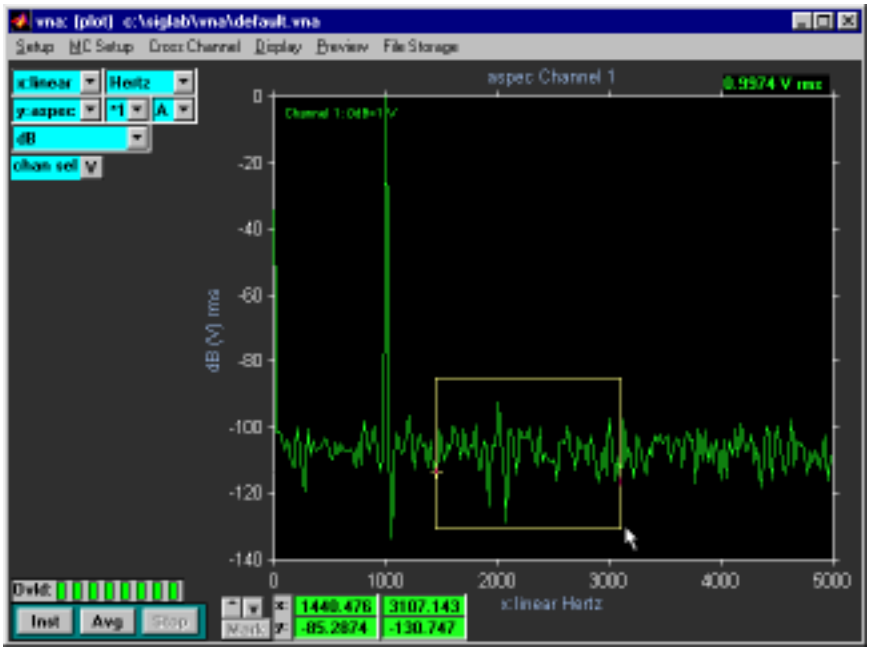


Figure 2-9. The yellow zoom box appears indicating the region of interest.

9. **Viewing Previous Expansions**. Make successive right single clicks in the plot window and you will see the plots scroll through the previous display expansions. Four different expansions are stored. When you hear a small click from your system speaker, this means you have cycled through the 4 expansions. The next click will autoscale the plot limits to the data as discussed in step 5 above.

10. **Cursoring Data**. You can cursor any point in the plot by left single clicking on it with the mouse. The corresponding x-y readout for that data point appears beneath the plot.

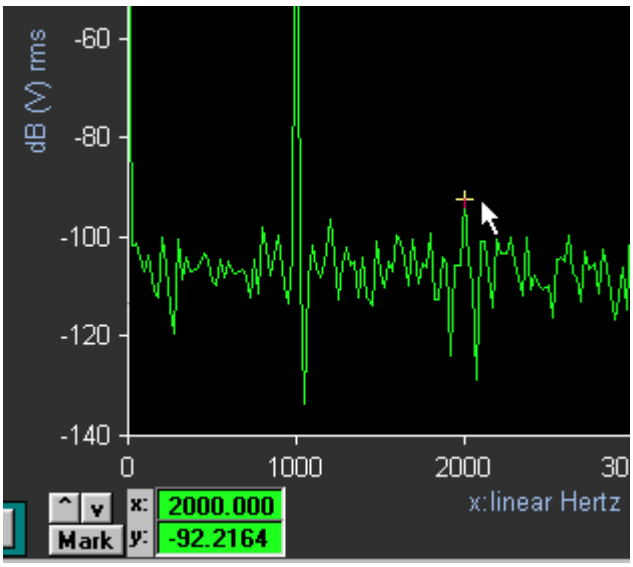
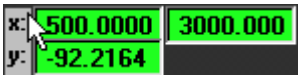


Figure 2-10. The x-y readout under each plot gives the current value of the cursored data point. In this figure the second harmonic of a 1.414 volt 1000 Hz sine wave was cursored. Notice that it's at 2000 Hz as expected and –92.2 dB down from the fundamental (1st harmonic).

11. **Setting the Plot Limits**. Most of the time, display expansion using the mouse is acceptable. However at times you'll want to set the plot x-y limits to some exact values. In other words, sometimes you want to set [Xmin Xmax] and [Ymin Ymax] explicitly. Here's how you do this.

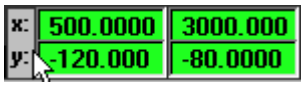
Clicking on **x:** allows entry of user-defined X limits



a. Set the X limits: **Left single click** on the **x:** button. A second edit box will appear next to the already present one. Enter Xmin and Xmax in the left and right hand edit boxes respectively.

b. Set the Y limits: **Left single click on the y: button**. Enter Ymin and Ymax in the left and right hand boxes respectively. As you enter the numbers you will a box drawn in the plot window marking the new limits.

Clicking on **y:** allows entry of user-defined Y limits



c. **Left single click** in the plot window to accept these new x-y limits. Verify the limits actually changed as in Fig. 2-11

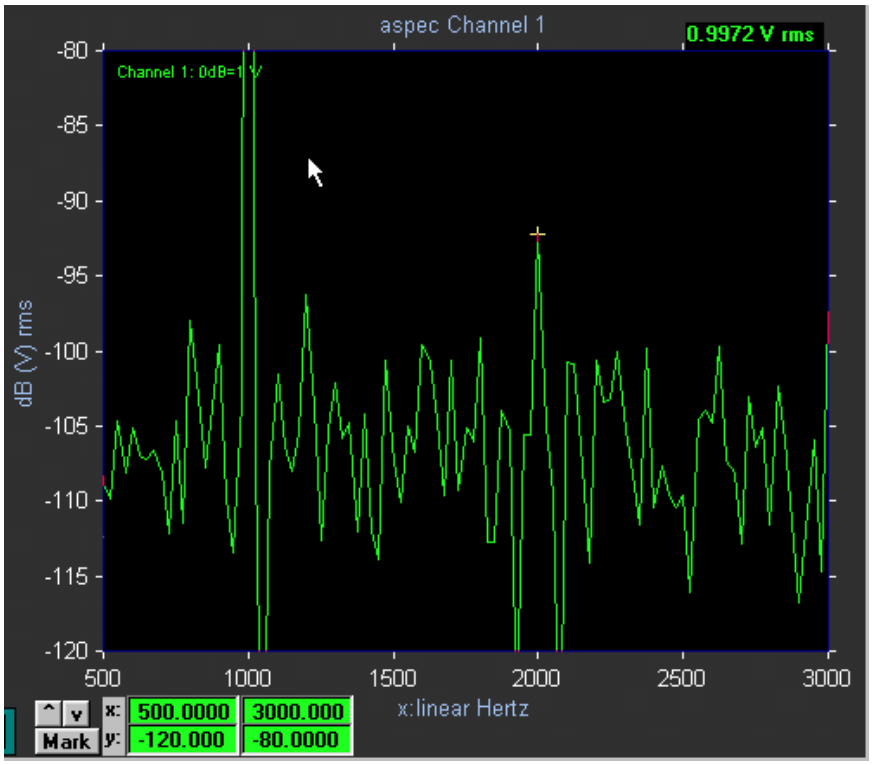
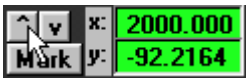


Figure 2-11. You can set x-y limits manually via the x: and y: controls beneath the plot. This plot extends from 500 Hz to 3000 Hz in the x direction and from –120 dB to –80 dB in y direction.

12. **Getting the Plot Limits**. At times you also may want to know what the current plot limits are. To do this, you can just look at the plot itself or you can **left single click on the x: and y:** buttons and the x-y limits will appear beneath the plot.

Clicking the ^ button finds successive peaks.



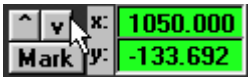
13. **Finding Peaks.** Frequently, you'll want to know where the largest signal or response resides in the spectrum. Sometimes your eyes will tell you but it's much easier to let the computer do comparisons for you. To do this, **left single click on the peak finder** beside the x: y: readout boxes.

    The peak finder appears like the "up carrat" symbol or Shift 6 ^ on most keyboards. Clicking on the peak finder again finds the second largest peak. Clicking again finds the third largest peak and so on until you wrap around to the largest peak again. Give it a try until you get the hang of it. Notice that each time you find a peak, the corresponding x-y peak values are readout beneath the plot.

14. **Finding Valleys.** Some people also call these valleys by a different name, "nulls" depending on the application. In any case, once you mastered finding peaks, finding

valleys is a snap. To find the lowest valley in your plot, **left single click on the v valley finder,** beside the x: y: readout boxes. It appear like a lower case letter v.

Clicking the v button finds successive valleys.

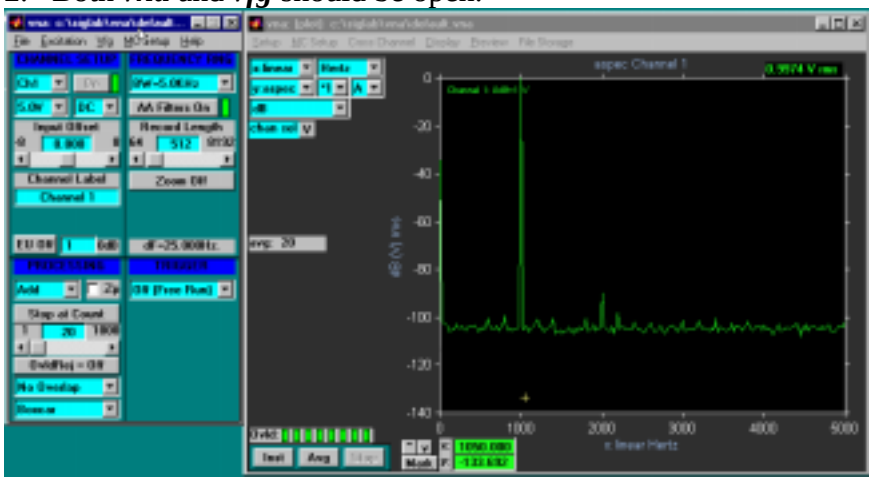


Clicking on the valley finder again finds the second lowest valley. Clicking again finds the third lowest and so on until you wrap around to the beginning. Give it a try until you get the hang of it. Notice that each time you find a valley, the corresponding x-y valley values are readout.

A Simple Exercise to Build On

Let's incorporate all we have learned so far into a simple exercise. This will give you some practice using the cursor functions, which can be a little awkward at first.

1. **Left single click on the Stop button** in the ***vna*** plot window if you haven't already done so.
2. Both ***vna*** and ***vfg*** should be open.



vna is set as follows:

1.Bandwidth = 5 kHz

2. Record Length = 512 samples

3. Channel 1 Full Scale = 5 volts

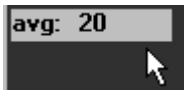
4. Boxcar window

5. Additive Averaging Stop Count = 20

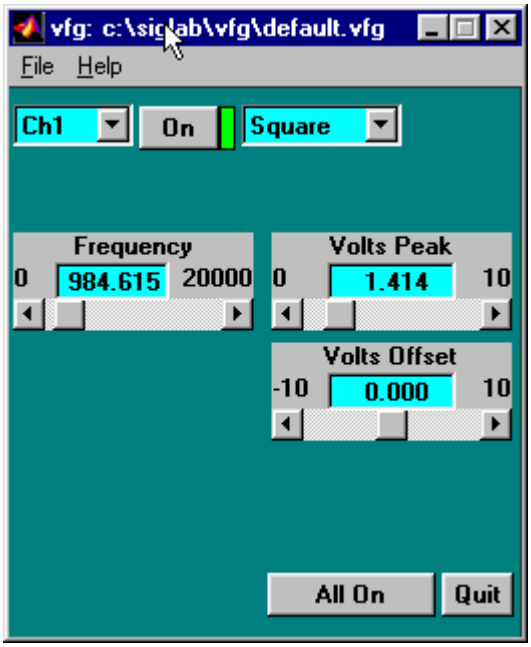
6. Autospectrum Single Display

3. Set ***vfg*** to output a **square wave on** channel 1. Use the default settings of 984 Hz and 1.414 volts peak.

4. **Hit the Avg** (average) button in the ***vna*** plot window. You will see the average counter count up from 1 to 20 in the ***vna*** plot window. You should see the first 3 harmonics of a



The average counter updates as the averaging is being performed



square wave, Figure 2-12. Remember to autoscale, **right click**, if the plot is not already autoscaled. Since the square wave harmonics do not lie on FFT lines, and the Boxcar window is being used, the noise floor is swamped by spectral leakage. Let's fix that.

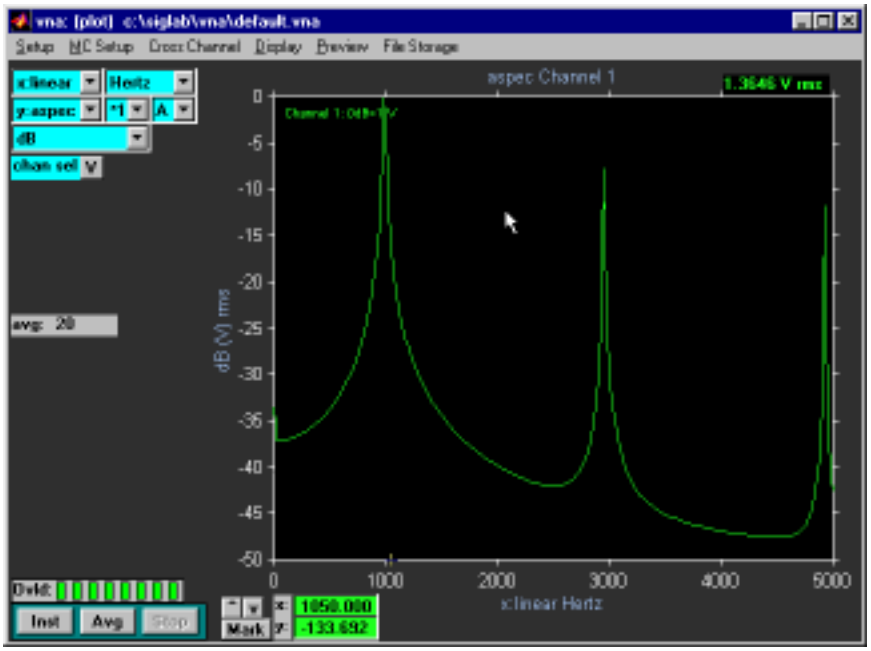


Figure 2-12. The first 3 harmonics of a 984 Hz square wave processed with a boxcar window. The dynamic range is greatly reduced by the spectral leakage of harmonic components into adjacent fft bins.

5. In the ***vna*** setup window, change the FFT window from Boxcar to **Hanning**. In the ***vna*** plot window menu, check **Display/Overlay Upper**. Hit **Avg** and observe the reduction in spectral leakage in Figure 2-13. Again, autoscale (right click) in the plot window if necessary.
6. In the ***vna*** plot window menu, **uncheck Display/Overlay Upper** such that you only see the most recent measurement.

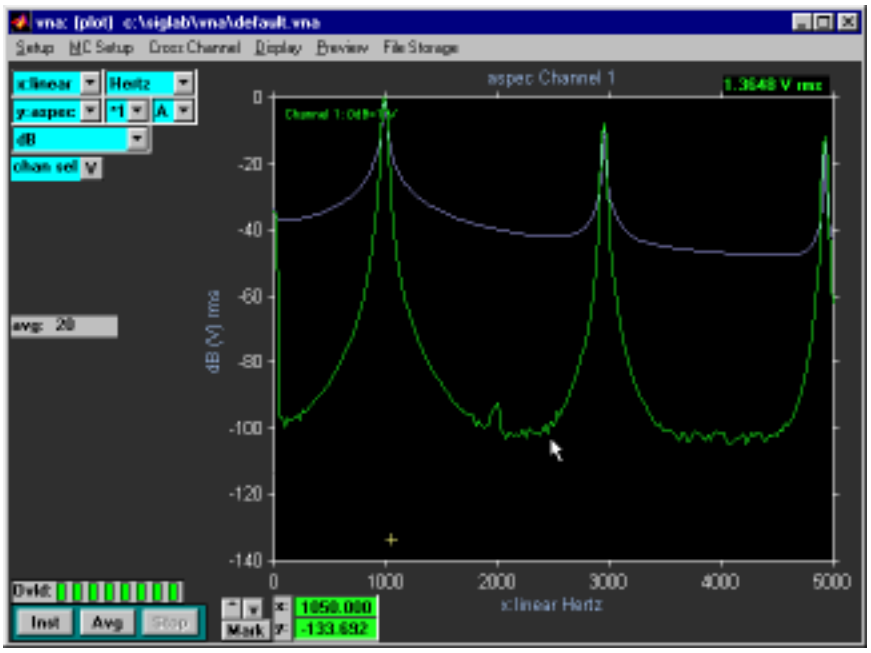


Figure 2-13. Though the fundamental, 3rd, and 5th harmonics of the 984 Hz square wave are clearly visible, there is substantial spectral leakage since they do not happen to lie exactly on FFT analysis lines.The dynamic range is greatly enhanced due to reduced spectral leakage. The mouse pointer points to the hanning windowed measurement.

The total RMS energy in the displayed portion of the signal is displayed above the plot in volts rms.

The RMS level meter updates in real-time as the signal level changes

7. Let's say you'd like to know the difference between the fundamental and the third harmonic in the spectrum. You could visually ascertain that the third harmonic is at approximately 3000 Hz and is about 10 dB down from the fundamental. However, here's how to get exact answers.

a. **Left single click anywhere in the plot window**.

Initialize mouse cursor in plot window

b. **Hit the peak finder button**. You should see the small yellow cross hair jump to the fundamental.

c. **Hit the mark button**. The Mark button hasn't been discussed up to this point. The small yellow cross-hair should turn into a large red "Marker" cross-hair. Now you have established a reference point. All subsequent cursored data points will be measured with respect to this x-y location.

x:	2950.000	1975.000
y:	-7.54674	-8.79076

Left boxes are absolute positions
Right boxes are relative positions

d. **Hit the peak finder button again**. You'll see the yellow cross-hair jump to the third harmonic and the x: y: readout will update as in Figure 2-14.

e. **Interpret the x: y: readout**. Four boxes appear next to the x: y: after you cursored the third harmonic. The left two boxes indicate the **absolute** position of the yellow cross-hair. The right boxes indicate the **relative** position with respect to the "marked" location.

f. **Click on the Mark button again to remove it**. The Mark button is a toggle button.

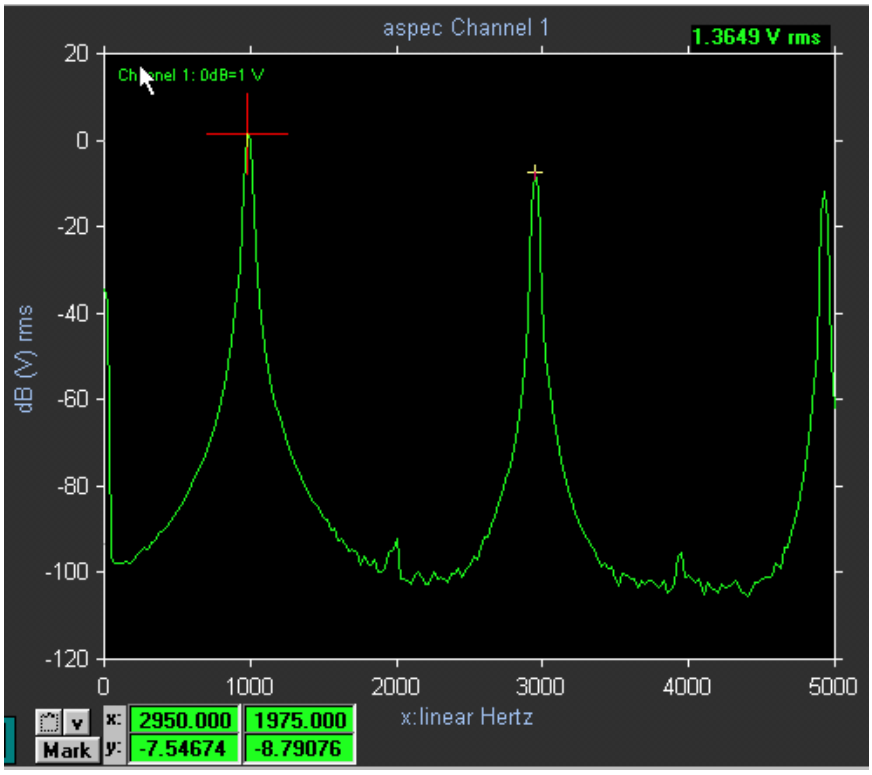


Figure 2-14. The difference between the fundamental and the third harmonic is measured using the Mark and peak finder buttons. The third harmonic is 1975 Hz to the right of the fundamental and 8.79 dB smaller.

9. You must be exhausted now. Let's save the work so we don't have to remember all these steps the next time we setup SigLab. **Select File/SaveAs** and save to the file **excercise1.*vna***. The data and the setup information are now contained in one concise file. It's not necessary to enter the ***.vna*** extension. It will be applied automatically.
10. Congratulations, you are done with the first Getting Started exercise. **Select File/Quit** in the ***vna*** setup window or click on the Windows 95/98/NT X button in the upper right hand corner to close ***vna*** and ***vfg***.

Post Processing

In this section we investigate the different options available for post-processing. There are two main approaches to processing your measurement data depending on your MATLAB programming level:

1. For the experienced MATLAB programmer,

   post process in the MATLAB environment using the SLm data structure.

2. For the non-MATLAB programmer,

   post-process in another environment like Excel using the data Export selection from the File menu.

Regardless of which approach you take, it's important to understand how SigLab stores data in the MATLAB environment.

Brief Architecture Overview

When you the push the **Inst** or **Avg** buttons in ***vna***, you're actually causing the following events to happen:

1. MATLAB requests new data from SigLab hardware via a DLL function call.
2. New data (if available) is returned to MATLAB **as a MATLAB variable**.
3. This MATLAB variable/vector is plotted in the ***vna*** plot window.
4. Goto step 1.

The above process continues until the user hits the Stop button.

Even though the data is returned as a MATLAB variable, it is hidden from the user by design. **Your data is a local variable to the vna.m M-file, not a global variable.** If you type "who" at the MATLAB prompt, you'll find no indication that a new MATLAB variable or variables exists after running ***vna***. In fact, hundreds of new variables actually do exist but they are all local variables. Why this approach? We hide the data for two reasons.

1. If it's not hidden, it's very easy to accidentally erase it or modify it by "hacking" in the MATLAB command window.
2. If it's not hidden, your MATLAB workspace would be filled with hundreds of meaningless variables that would leave you either very perplexed or very angry at the vendor.

Post-Processing with MATLAB

There exists a single data access method. It is a function call of the following form:

```
SLm = vna('get','meas')
```

SLm is a MATLAB data structure containing measurement related information including the data itself. The `vna('get','meas')` command extracts the most recent measurement data from ***vna***'s local variable space. The SLm data structure contains the data listed in Table 1. For a complete description of how to use SLm, see Section 2.0 of the Programming Guide.

SLm fields	**Description**
Scmeas: [1x16 struct]	Single-channel Measurement Information
Xcmeas: [4x16 struct]	Cross-channel Measurement Information
Xcstate: [1x1 struct]	Cross-channel State Information
Tdxvec: [1x512 double]	Time-domain x Vector
Fdxvec: [1x201 double]	Frequency-domain x vector
clist: [1 2]	Channel List of Enabled Channels
numin: 4	Number of Channels in Current Configuration
Wincor: 1	FFT Window Correction Factor
Winsel: 1	FFT Window Index
rbw: 25	FFT Resolution Bandwidth
navg: 0	Number of Averages
zpad: 0	Zero Padding On/Off
ovld: 0	Overload Status
Zoomcf: 0	Zoom Center Frequency
Filestor: [1x1 struct]	File Storage Menu Information

Table 2-2. The SLm data structure

A Brief SLm example

Although the SLm section does an exhaustive job of explaining it contents and uses, a few quick examples should serve to get you over the hump of using SLm. The following steps were performed from the MATLAB command window. You'll find yourself understanding how the SLm data structure works if you go through these examples. Afterwards, please review Section 2.0 of the Programming Guide.

Open ***vna*** if it already is not.

>> `vna` <enter>

Data is now available in the MATLAB environment although you can't see it as discussed above. The data is local to the vna M-file. To access the data, namely the SLm data structure, use the data access method listed above. The `vna('get','meas')` command extracts the most recent measurement data from ***vna***'s local variable space. What happens if you haven't made a measurement yet? Do we have any measurement data? The answer is Yes! In this case, ***vna***'s local measurement variables have already been set by reading from the current setup file, in this case, default.vna.

Step 1:

>> SLm = vna('get','meas');

If you accidentally omit the trailing semi-colon, you'll quickly understand why it's commonly used. Omitting the semi-colon is tantamount to saying, "Show me the data"!

Step 2:

To see all the fields of SLm, do this:

>> SLm

SLm =

scmeas: [1x16 struct]

xcmeas: [4x16 struct]

xcstate: [1x1 struct]

tdxvec: [1x512 double]

fdxvec: [1x201 double]

clist: [1 2]

numin: 8

wincor: 1

winsel: 1

rbw: 25

navg: 20

```
      zpad: 0
      ovld: 0
    zoomcf: 0
  filestor: [1x1 struct]
```

The first field listed above in *SLm* is *scmeas*. scmeas contains single channel measurement data. Here, we will investigate both *scmeas* and *tdxvec*. *scmeas* is also a structure. Essentially, *scmeas* is also structure. More precisely, scmeas is an array of structures. On the other hand, *tdxvec* is a vector. This will all become clear after we look at some examples

Step 3.

To get a quick plot of channel 1 time domain data, do this:

```
>> figure;
>> channel = 1;
>> x = SLm.tdxvec;
>> y = SLm.scmeas(channel).tdmeas;
>> plot(x,y);
```

After typing SLm.scmeas, it reported a **1x16 struct array with fields.** If you are familiar with C language, this is called an **array of structures.** Each element of the array is a data structure containing that channel's single channel measurement data.

Step 4.

To make the concept of an array of structures more clear, let's look at one element in the array. First, let's look at channel 1's single channel measurment data:

```
>> SLm.scmeas(1)

ans =

      tdmeas: [512x1 double]
       aspec: [201x1 double]
         fft: []
        acor: []
       label: 'Channel 1'
    eu_on_off: 0
  euscale_fac: 1
    eu_string: 'Gs'
```

eu_val: 1

fs_val: 5

db_ref: 1

Note that whereas SLm.scmeas is an array of structures, SLm.scmeas(1) is just a structure or one element in the array. Equivalently, we can type:

>> SLm.scmeas(2) … to see channel 2 data

>> SLm.scmeas(3) … to see channel 3 data

…

>> SLm.scmeas(16) … to see channel 16 data

>> SLm.scmeas(17)

??? Index exceeds matrix dimensions.

SLm.scmeas(17) is invalid since SigLab only supports up to 16 input channels at the time of this writing.

Step 5

Extract the channel 1 time data and assign it to the MATLAB variable y:

>> y = SLm.scmeas(1).tdmeas;

It's that simple. Now extract the time domain x vector associated with the time data:

>> x = SLm.tdxvec;

If you understand what we have done so far, then 90% of the battle is over.

Step 6.

Let's go through a complete exercise together.

To get a quick plot of channel 1 time domain data, do this:

>> figure;

>> channel = 1;

>> x = SLm.tdxvec;

>> y = SLm.scmeas(channel).tdmeas;

>> plot(x,y);

Step 7.

To get a quick plot of the channel 1 autospectrum data, do this:

>> x = SLm.fdxvec;

>> y = 10*log10(abs(SLm.scmeas(channel).aspec)));

>> plot(x,y)

SLm.fdxvec stands for the frequency domain x vector associated with autospectrums.

For practice, make a new measurement using ***vna*** and repeat the above steps. Verify the plots reflect your new measurement.

Post-Processing from a Setup File

vna doesn't have to be open as a pre-requisite for post processing. If you saved your results to a setup file like default.vna, then you can extract that data on a machine without installing SigLab software. Here's how you do it. From the MATLAB Command Window,

>> load default.vna –mat

Now SLm exists in the MATLAB workspace for post-processing as before. To verify this type:

>> who

Your variables are:

CenterFreq	ch_ptr	vdlg1_s1
ChanLabel	exdlg2_s1	vdlg1_s2
ChanStat	exdlg2_vis	vdlg2_s1
Cmprssd_Notes	grids	vi_timestamp
EULabel	hdlg1_s1	vna_pos
SLm	hdlg2_s1	xplot_axes
SampleRate	hdlg2_vis	xplot_s1
SystemClk	key	xplot_s2
UniformFlg	num_io	

Ignore the other variables as they are internal ***vna*** file details. The only variable of interest is SLm. You can repeat steps 1 through 7 using SLm.

Two-Channel Measurements

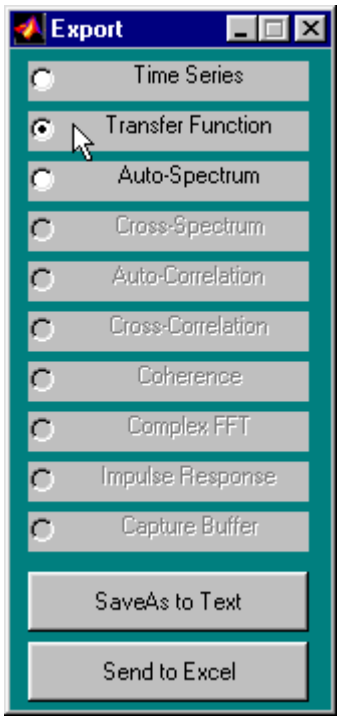
Post-processing two-channel measurements is a more involved task and as such needs more explanation. Refer to SLm section (Section 2.0) of the Programming Guide for how to extract and plot transfer function, cross-spectrum, correlation, and coherence data.

Post-Processing Outside of MATLAB

You may have already developed a set of routines for post-processing data independent of MATLAB. In that case, you'll need to export the data to your environment. SigLab exports data by way of ascii text files. For Excel users, we have developed a set of templates and Visual Basic scripts optimized for processing these ascii files.

To export your data to an ascii text file:

1. Select the *Export* menu pick under *File* in the ***vna*** setup window.
2. Select the data type of interest.
3. Click *SaveAs* and indicate the filename.



To export your data to Excel:

1. Select the Export menu pick under File in the ***vna*** setup window.
2. Check the data type of interest.
3. Click Send to Excel. When you do this, a text file is automatically created. The text file name is the same as the currently opened ***vna*** file but with a .txt extension.

How to customize the Excel interface is described in the Common Element of Section (Section 4.0) of this User Guide.

A Brief Review of the VNA Setup Window

This section gives a quick introduction to all the controls in the ***vna*** setup window. For an in-depth explanation, see the vna section of the user manual. The ***vna*** setup window is divided into 4 control sections or dialogs as they're frequently referred to.

Channel Dialog

For those familiar with using a stand-alone oscilloscope, channel controls are similar to vertical controls. Only one channel's setup is viewable at any give time. First, we'll look at Channel 1 only. To view channel setup information at a glance use the MC (Multi-Channel) Setup menu selection in the ***vna*** Setup window.

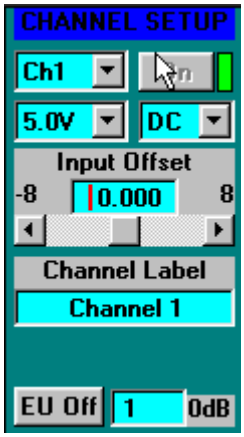


Figure 2-16. A sample channel setup dialog found in the VNA setup window.

Find the following settings in Figure 2-16.

A. **Channel 1 is ON**. Note that Channel 1 cannot be turned off. All others channels can be turned off.

B. **Full Scale Voltage Range = ±5 volts peak**. All voltage ranges are implied bipolar(±) although not listed that way in the user interface. You can set the full scale range from ±20 mv to ±10 volts peak.

C. **Coupling = DC**. Your other choices are AC and ICP.

D. **DC Offset = 0 volts**. The range of offset values depends on the full scale voltage range.

Full Scale Voltage Range	**Maximum DC Offset**
± 10.0	± 8.0 volts
± 5.0	± 8.0 volts
± 2.5 volt to ± 20 mv scales	± 2.5 volts

E. **Channel Label = "Channel 1"**. You can edit the channel label to read any text string. If the string is too long, it will scroll to the left.

F. **EU (engineering units) are Off**. This toggle button controls the state of engineering units. When engineering units are off, the units are based on volts. When engineering units are on, the units are based on your entries. See Figure 2-18.

G. **0 dB reference = 1 volt RMS**. You can changed the 0 dB reference point to any value. The default value is 0 dB = 1.0 volt RMS. Refer to the VNA section on engineering units for a complete description of this field's capabilities.

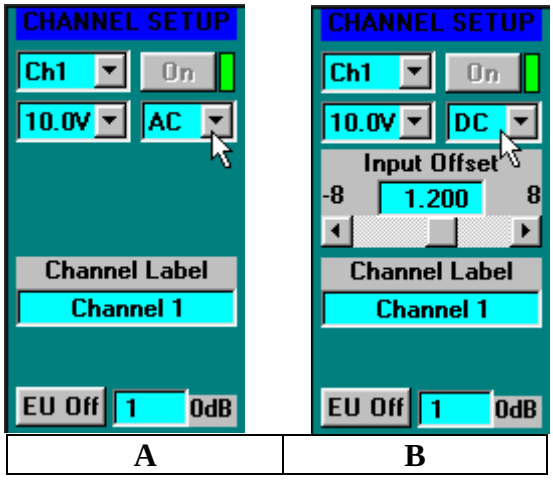


Figure 2-17. Two faces of the Channel Setup dialog, with AC coupling (A) and DC coupling (B). With AC coupling, the DC offset slider is not visible. With DC coupling, a slider appears for selecting the DC offset

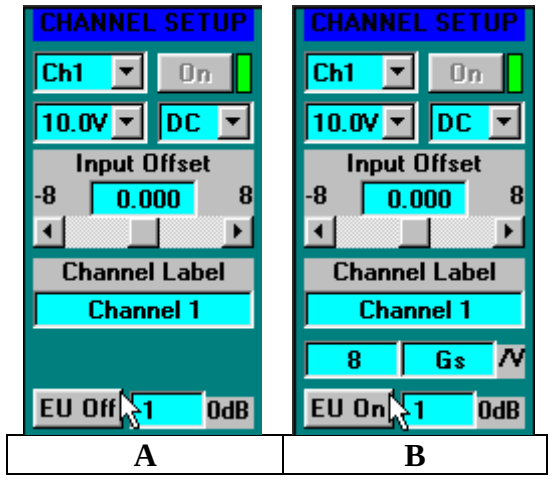


Figure 2-18. Two faces of the Channel Setup dialog, without engineering units (A) and with engineering units (B). With EU Off, the two fields to enter engineering units are not visible. With EU On, you have to enter both an engineering unit and multiplying factor.

MC Setup for Multi-Channel Users

For SigLab users in the 4 to 16 channel range, using the Channel Setup dialog can be quite tedious for both channel setup and interrogation. To make life easier, we have provided a Multi-Channel Setup utility, MC Setup. It has all the functionality of the Channel Setup dialog but it lets you see all channels at a glance. You'll find a main menu link to MC Setup in both the VNA Setup and Plot Windows.

Figure 2-19. The VNA plot window has a link to the MC Setup window.

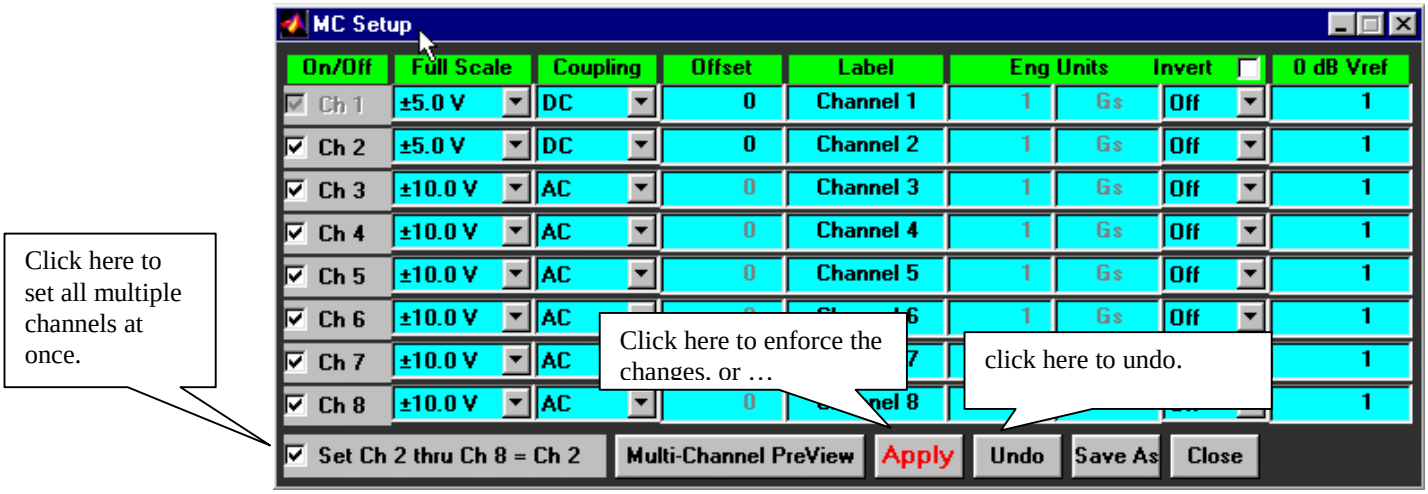

MC Setup
On/Off
Full Scale
Coupling
Offset
Label
Eng Units
Invert
0 dB Vref
Ch 1 ±5.0 V DC 0 Channel 1 1 Gs Off 1
Ch 2 ±5.0 V DC 0 Channel 2 1 Gs Off 1
Ch 3 ±10.0 V AC 0 Channel 3 1 Gs Off 1
Ch 4 ±10.0 V AC 0 Channel 4 1 Gs Off 1
Ch 5 ±10.0 V AC 0 Channel 5 1 Gs Off 1
Ch 6 ±10.0 V AC 1 Gs Off 1
Ch 7 ±10.0 V AC 1
Ch 8 ±10.0 V AC 0 1
Click here to set all multiple channels at once.
Click here to enforce the changes. or …
click here to undo.
Set Ch 2 thru Ch 8 = Ch 2
Multi-Channel PreView
Apply
Undo
Save As
Close

Figure 2-20. The multi-channel setup window as called from VNA.

Key points to be aware of when using the MC Setup Window

1. No changes take affect until the Apply button is pushed. The Apply button is disabled until you change some parameter in the MC Setup window.
2. After you push the Apply button you cannot Undo the changes. The Undo button only undoes changes since the last Apply.
3. Channel 1 is always on. It cannot be disabled.
4. Channels 2 thru N can be turned on at once as indicated in Figure 20.
5. The Invert checkbox is a display feature that applies to the Engineering Units fields only. If unchecked, the engineering units are displayed as EU per Volt. If checked, they are displayed as Volts per EU.

Frequency Dialog

The frequency controls are similar to the so-called horizontal controls on stand-alone oscilloscopes. The frequency controls impact your entire system, not just a particular channel or set of channels. In other words, you can't set the bandwidth on channel 1 to 20 kHz and the bandwidth on channel 2 to something different like 100 Hz. All channels must acquire data at the same bandwidth and the same frequency resolution.

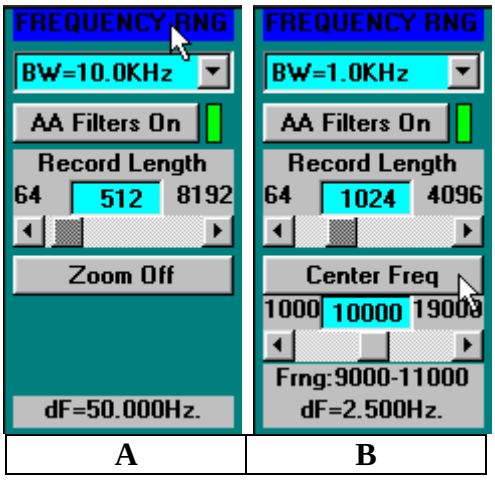


Figure 2-21. Sample Frequency Range Dialogs. Part A is setup for baseband analysis. Part B is setup for zoom analysis with the center set to 10 kHz and a bandwidth of ±1.0 kHz.

Refer to Figure 2-21, part A for the following settings:

A. **BW (Bandwidth) = 10 kHz**. The sample rate is always 2.56 * Bandwidth. In this case, Fs = 25.6 kHz. The range of bandwidths is 2 Hz to 20 kHz for a 20 kHz system.
B. **AA (Anti-Alias) Filters are On**. AA Filters may be toggled on and off. Only the digital AA filters may be toggled on/off. The 20 kHz AA filter is analog and is always on.
C. **Record Length = 512 samples**. For spectral analysis, the record length is equivalent to the FFT size which is always a power of 2. The range of FFT sizes is 64 to 8192.
D. **Zoom Processing is Off**. When Zoom is off, we are said to be "processing in baseband". Zoom processing is equivalent to shifting the center frequency of analysis from 0 Hz (baseband) to some center frequency of interest.
E. **dF = 50 Hz**. The FFT bin width = 50 Hz. The terms dF, FFT bin width, and frequency resolution all refer to the same thing. In generic terms,
   dF = Sample_rate/Record_Length.
   For our current setup, dF = 25,600 Hz/512 = 50 Hz.

Processing Dialog

Processing controls are less associated with oscilloscopes than they are with spectrum analyzers or network analyzers. Unlike the channel and frequency controls which affect the way the analog data is acquired, the processing controls manipulate the data only after it has been digitized.

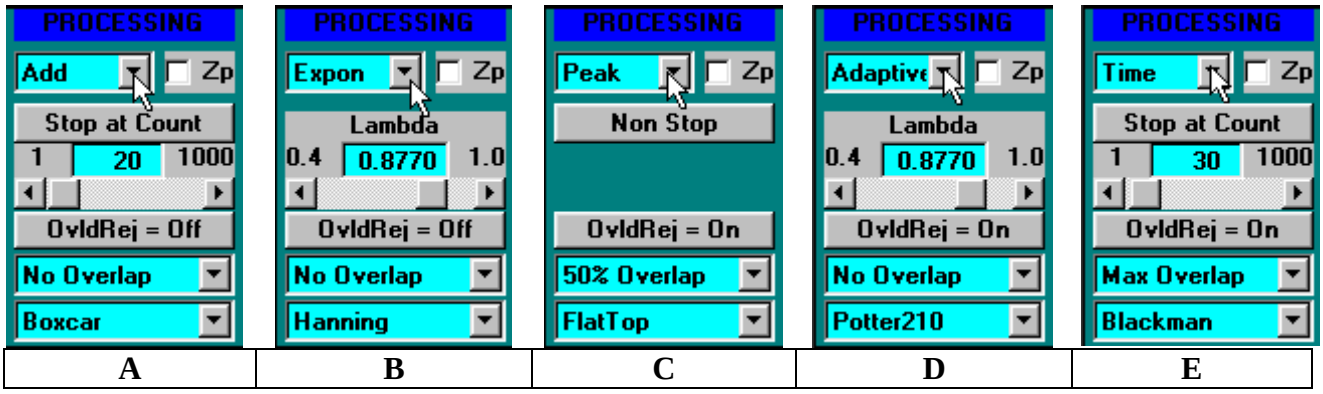


Figure 2-22. The many different faces of the Processing Dialog. Each dialog shows different configurations for averaging type, overload rejection, % overlap, and FFT window.

Refer to Figure 2-22, part A for the following settings:

A. **Additive Averaging**. In the processing control you will only see Add to signify additive averaging. See the VNA section of the user's manual for a complete explanation and the equations used in the different averaging modes.
B. **Zero Padding (Zp) is Off**. Zero padding is generally only set to On for correlation measurements.
C. **Stop at Count = 20**. The acquisition and averaging will automatically stop after 20 averages are complete.
D. **Overload Reject is Off**. In this state, even overloaded data (i.e. saturated, corrupted, bogus data) is included in the averaging process. If Overload Reject is on then overloaded frames of data are not included.
E. **No Overlap**. You can set the % overlap to be No Overlap (0%), 50% Overlap, or Max Overlap (93.75%).
F. **Boxcar Window**. You can choose numerous FFT analysis windows to control spectral leakage. The boxcar window is equivalent to no window at all.

Triggering Dialog

The triggering controls are more relevant to time domain measurements and less relevant to spectrum measurements. Most spectrum measurements don't require triggering. If this is the case, set triggering to **Off (Free Run)**. You may only see **Off (Free Ru** depending on your font size and type settings.

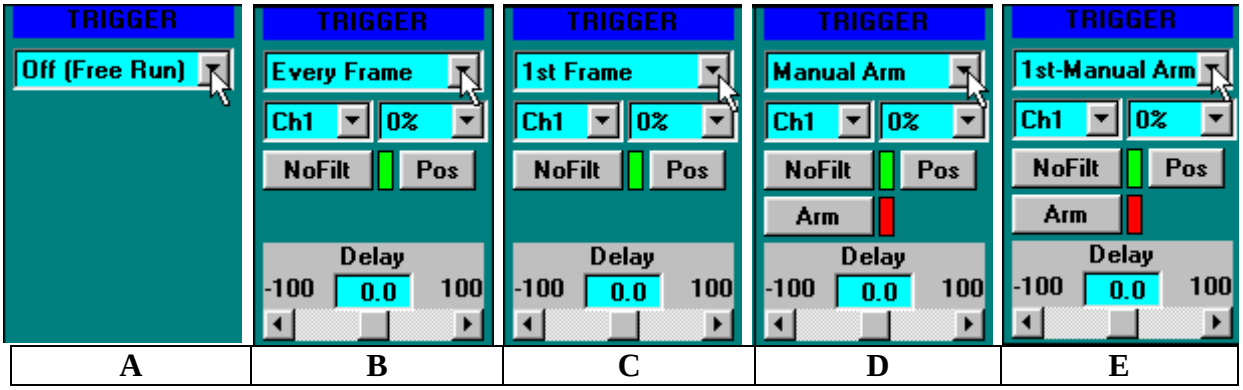


Figure 2-23. The many different triggering modes from A to E. Only the controls relevant to your particular triggering mode are displayed. This helps avoid confusion.

Trigger mode A is Free Run and is equivalent to no triggering at all. The data is displayed as it is acquired in Record Length blocks.

Trigger mode B is Every Frame triggering. Data is displayed (triggered) in Record Length blocks every time a positive-going signal crosses through 0% of full scale on channel 1.

Trigger mode C is 1st Frame triggering. It is equivalent to Trigger mode B (Every Frame) on the first block of data and Trigger mode A (Free Run) thereafter.

Trigger mode D is Manual Arm triggering. It is equivalent to Trigger mode B except that you must arm (press) the Arm button before trigger events are monitored. The trigger is "armed" when the arm led is green.

Trigger mode E is 1st Frame Manual Arm triggering. It is equivalent to Trigger mode C except that you must arm (press) the Arm button before trigger events are monitored.